
¿Qué es el software libre?

Jesús M. González Barahona

jgb@debian.org

jgb@gsync.escet.urjc.es



Consol 2003, México, 8 de febrero de 2003



¿Por qué hablar de software libre?

Desde hace 30 años, nos hemos acostumbrado a que:

- El productor puede imponer en qué condiciones se puede usar su producto
- Es normal que no se regulen nichos de mercado básicos para la economía con situaciones de monopolio casi absoluto
- Si se detectan errores en un producto, sólo el productor puede arreglarlos
- No tiene sentido adaptar un producto a necesidades particulares (más allá de lo que se haya previsto)
- Copiar información es “malo”

¿Qué es software libre?

Quien lo recibe puede (pero no está obligado a):

- usarlo como mejor le parezca, donde mejor le parezca.
- redistribuirlo a quien quiera, por los medios que quiera.
- modificarlo (y mejorarlo o adaptarlo).
- redistribuir las modificaciones

Imprescindible: disponibilidad de código fuente.

software libre \neq software gratis

<http://www.fsf.org/philosophy/free-sw.es.html>

<http://opensource.org/docs/osd-spanish.html>

¿Y por qué es esto y no otra cosa?

Desde luego no es casualidad...:

- Motivos éticos: porque las cosas deberían ser así.
- Motivos prácticos: porque las cosas funcionan mejor así.

Largas discusiones, que han asentado cierto consenso:

- Debian Free Software Guidelines,
- Definición de “Open Source”.
- Definición de software libre de la FSF

La ética del programador

- Un buen programador debería contribuir con su trabajo a la Comunidad.
- Un buen programador debería poder aprovechar el trabajo de otros buenos programadores.
- Un buen programador debería poder “arreglar” y mejorar cualquier programa.
- Un buen programador se siente orgulloso de usar su código, y de que otros lo usen.

Buen programador = Hacker

Ideas formuladas por Richard Stallman, continuadas por la FSF, la comunidad BSD, y otros.

¿Y los argumentos prácticos?

- Nuevos modelos de desarrollo (bazar frente a catedral).
- Ventajas del escrutinio público y de la mejorabilidad.
- Competencia real en el desarrollo y el mantenimiento.
- Viabilidad técnica frente a mercadotecnia.
- Nuevas posibilidades de negocio (ej: desafío a posiciones de monopolio).

Ideas formuladas por Eric Raimond, promovidas por la Open Source Initiative y otros.

Consecuencias de la “libertad” del software

- **Coste:** modelo de costes radicalmente distinto al del software propietario.
- **Apertura:** puede modificarse, puede inspeccionarse, puede estudiarse.
- **Distribución:** nuevos canales, nuevos métodos.
- **Desarrollo:** modelos de desarrollo “sorprendentes”.
- **Mantenimiento y soporte:** Verdadera competencia.

Combinación de dos poderosos mecanismos:

- Competencia (pudiendo usar el mismo programa base)
- Cooperación (incluso involuntaria)

¿Va a ser el software libre una revolución como la de Internet?

¿Y por qué me interesa todo esto?

- La visión del usuario final (individual o empresa).
- La visión del desarrollador (o productor de software).
- La visión del integrador.
- La visión del que proporciona mantenimiento y servicios.

El usuario final

Están los “olvídate”...

- Olvídate de los monopolios (verdadera competencia, mejores productos, mejores servicios).
- Olvídate de la “fiabilidad” del productor (el futuro lo asegura la aceptación del producto, y la disposición del fuente).
- Olvídate de tomar decisiones con pocos elementos (puedes probar el software en su entorno real a coste prácticamente cero).
- Olvídate de depender de la estrategia de tus proveedores.
- Olvídate de “cajas negras” en las que has de confiar “porque sí”

El usuario final

...y los “¿qué tal si...?”

- ¿Qué tal si pudieras adaptar/personalizar el producto como quieras?
- ¿Qué tal si pusieras “estar a la última” a bajo coste?.
- ¿Qué tal si pudieras arreglar los problemas (o pagar para que los arreglen)?
- ¿Qué tal si pudieras decidir sobre la evolución futura del producto?
- ¿Qué tal si pudieras contratar la integración de los dos mejores productos en el entorno que te interesa?
- ¿Qué tal si pudieras contratar auditorías completas de cada producto por terceras partes?

El usuario final

Gran parte del control pasa al usuario
(frente al productor de software).

El desarrollador/productor de software

El software libre cambia las reglas del juego.

- Puedes competir siendo pequeño.
- Es mucho más fácil adquirir tecnología punta (y más barato).
- Te puedes aprovechar del trabajo de tu competencia (ojo: también tu competencia del tuyo).
- Si lo haces bien, puedes conseguir, a bajo coste, la colaboración de mucha gente.
- El canal de distribución es mucho más barato, y global.
- Es posible convertirte en aplicación de referencia mucho más fácil.

El desarrollador/productor de software

¿Y de dónde saco el dinero?

- El mejor conocimiento sobre el programa lo tiene su desarrollador.
- Si se cuida la imagen, el desarrollador es el “punto más visible”.
- Desarrollos a medida, modificaciones, adaptaciones.
- Soporte “a lo grande” (corrección de erratas, acceso preferente a nuevas versiones, nuevas características, etc.)

Si hay gente que quiere software, y está dispuesto a pagarlo, algún desarrollador/productor se beneficiará...

El integrador

¡Bienvenido al paraíso!

- Todos los productos libres están a tu disposición (¡y sin preocuparte de licencias propietarias!).
- Si los productos no “encajan”, puedes “limarlos” (tienes el código fuente, puedes conseguir interoperabilidad).
- Puedes integrar “trozos” de productos, o productos enteros, o lo que sea.
- No más cajas negras: las tripas de todo son transparentes.

Puedes construir sobre el trabajo de otros, en igualdad de condiciones con esos otros.

Mantenimiento y servicios

El disponer del fuente lo cambia todo.

- Estás en las mismas condiciones que el productor.
- Competencia en el negocio del mantenimiento.
- El valor añadido de los servicios es mucho más apreciado (el coste del programa es bajo).
- El conocimiento del estado del arte es muy importante (es bueno tener relación con los proyectos libres).
- Negocios nuevos: consejo sobre versiones y combinaciones de programas, información sobre nuevos desarrollos, gestión de proyectos libres.
- Este es actualmente el negocio más claro.

Principales obstáculos para su adopción

- Técnicas FUD (miedo, desconocimiento, duda)
- “Disolución” (sistemas que pueden confundirse con el software libre)
- Desconocimiento (pérdida de visión): ¿por qué es interesante el software libre? (no quedarse sólo en el coste por licencia)
- Impedimentos legales: por ejemplo, patentes software.

Y habrá más...

¿Hay conclusiones?

- El software libre muestra ser un modelo económica y técnicamente viable.
- Detrás de él hay motivaciones técnicas, económicas y éticas.
- Es muy importante conocer el mundo en que nos movemos...
- El futuro depende, en gran parte de nosotros (como profesionales, como clientes, como empresarios,...).

Este es uno de esos raros momentos en los que toda una industria puede estar cambiando de paradigma.

Algunas URLs

- Grupo de trabajo de la Comisión Europea sobre software libre:
<http://eu.conecta.it>
- Open Sources (O'Reilly)
<http://www.oreilly.com/catalog/opensources/>
- Curso de software libre:
<http://curso-sobre.berlios.de>
- Zope Insider's Guide:
<http://www.opticality.com/Press/ZopeCorp/ZIG>
- Setting up shop
<http://www.openresources.com/documents/setting-up-shop/>